ГЛАВА XX.
Случайные леса

Когда ты в лесу,
ты становишься частью леса
Х. Мураками

Как можно заблудиться в лесу?
Там же все деревья разные!
О. Громыко

Случайный лес (random forest) – это бэгинг над решающими деревьями, при построении которых для определения каждого расщепления просматривается лишь подмножество всех признаков. В задаче регрессии ответы отдельных деревьев усредняются, в задаче классификации принимается решение голосованием по большинству[1]. Для построения ансамбля строим заранее заданное число – n_estimators – деревьев. Все деревья строятся независимо по следующей схеме, для каждого дерева:

- Выбирается подвыборка обучающей выборки размера max_samples (в классическом варианте с возвращением – с помощью бутстрепа) – по ней строится дерево.
- Для построения каждого расщепления в дереве просматриваем max_features случайных признаков (для каждого нового расщепления – свои случайные признаки).
- Выбираем наилучшие признак и расщепление по нему (по заранее заданному критерию[2]).

Каждое дерево строится, как правило, до исчерпания выборки (пока в листьях не останутся представители только одного класса), но в современных реализациях есть гиперпараметры, которые ограничивают высоту дерева, число объектов в листьях и число объектов в подвыборке, при котором проводится расщепление.

[1] В библиотеке scikit-learn в реализации классификатора RandomForestClassifier усредняются вероятности принадлежности классам и выбирается класс с максимальной вероятностью.

[2] Чаще – как принято при построении деревьев: используют энтропийный или джини в классификации и выборочную дисперсию в регрессии.

На рис. XX.01 показаны разные деревья из случайного леса и бустреп-подвыборки, по которым они были построены. На рис. XX.02 в этой же задаче показана полная обучающая выборка и случайный лес при различном числе деревьев. Видно, что общий вид разделяющей поверхности достаточно быстро формируется, но она не выглядит «достаточно гладкой», как, например, в методе kernel SVM. Разделение с помощью отдельных деревьев содержит артефакты – узкие области. Заметим, что каждое отдельное дерево настроилось на свою выборку (разделяет объекты со 100%-м качеством), при этом разделение с помощью леса почти не содержит артефактов (в данном случае – также при 100%-м качестве разделения выборки).

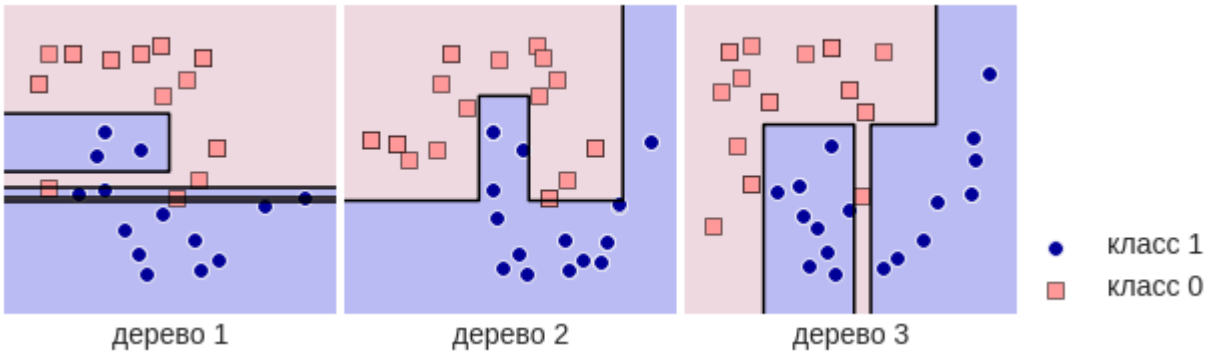

Рис. XX.01. Разные деревья в случайном лесе.

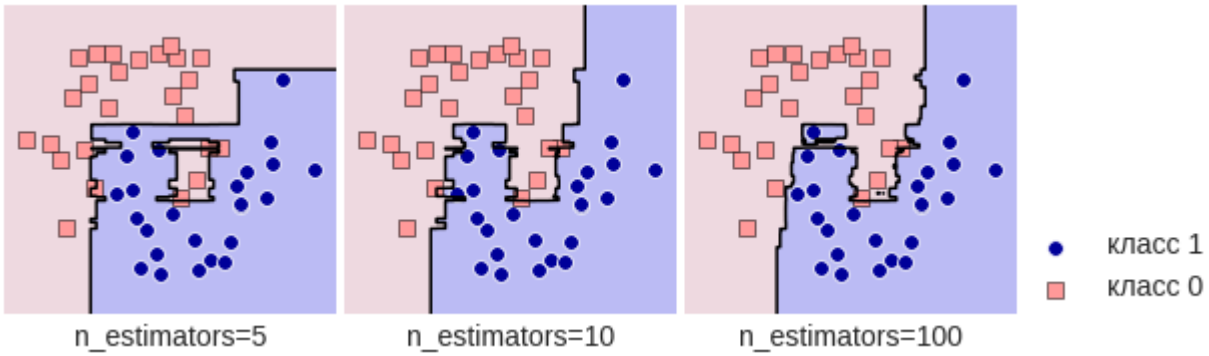

Рис. XX.02. Случайный лес с 10 (слева), 100 (центр), 1000 (справа) деревьями.

Рассмотрим теперь наиболее важные гиперпараметры метода, опишем как от них зависит качество решения задачи случайным лесом.

1. Число деревьев – n_estimators

Чем больше деревьев, тем не хуже качество случайного леса, но время обучения и работы ансамбля также пропорционально увеличиваются. При увеличении гиперпараметра n_estimators качество на обучающей выборке не уменьшается (может даже доходить до 100%), а качество на тесте как правило меньше (здесь есть переобучение), но тоже почти монотонно и выходит на

асимптоту (можно оценить, сколько деревьев достаточно для достижения максимального качества), см. рис. XX.03

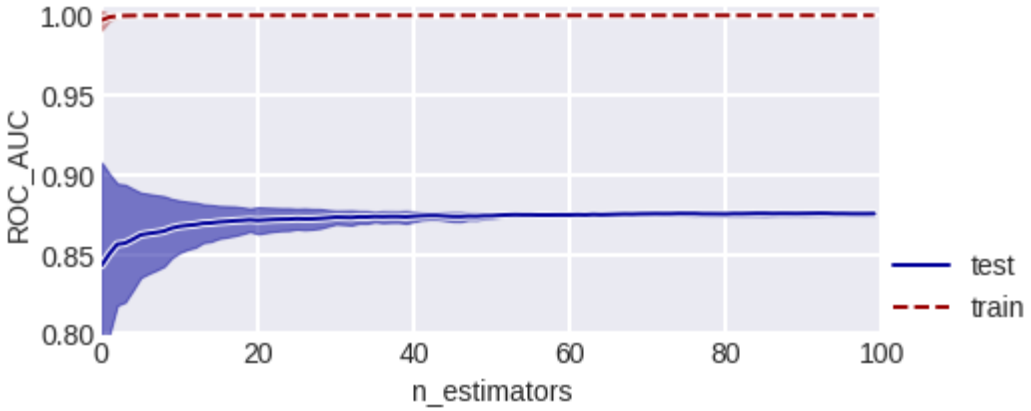

Рис. XX.03. Изменение числа деревьев в лесе.

2. Число признаков для выбора расщепления – max_features

График качества на тесте от значения этого гиперпараметра, как правило, унимодальный – поэтому необходимо найти его максимум. На обучении качество возрастает при увеличении max_features, т.к. использование всех признаков позволяет лучше настроиться на выборку. При увеличении max_features увеличивается время построения леса, а деревья становятся «более однообразными», например увеличивается корреляция между их ответами, см. рис. XX.05. По умолчанию этот гиперпараметр равен

Самый важный гиперпараметр RF – число признаков для выбора расщепления.

$\sqrt{n}$ – в задачах классификации,

$n/3$ – в задачах регрессии[1],

n – число признаков в задаче (хотя часто это не оптимальный выбор). Это самый важный гиперпараметр[2]! Его настраивают в первую очередь (при достаточном числе деревьев в лесе). Заметим, что при $\text{max\_features} = n$ случайный лес превращается в бэгинг над деревьями. Также отметим, что выполнение неравенства $\text{max\_features} < n$ не означает, что дерево зависит не от всех признаков (просто при выборе каждого расщепления просматриваются не все).

[1] В последних версиях библиотеки sklearn по умолчанию max_features равен числу признаков в задаче регрессии.

[2] При ансамблировании лесов логично строить леса с разным значением гиперпараметра max_features.

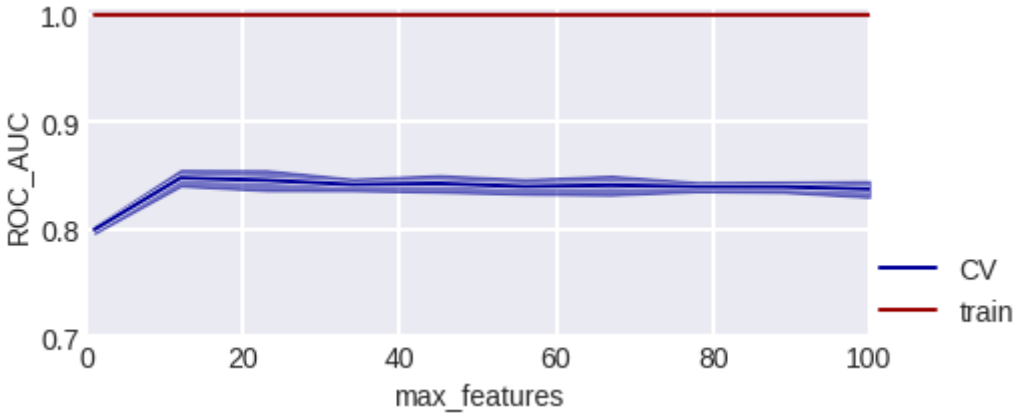

Рис. XX.04. Качество при разном числе признаков для выбора расщепления.

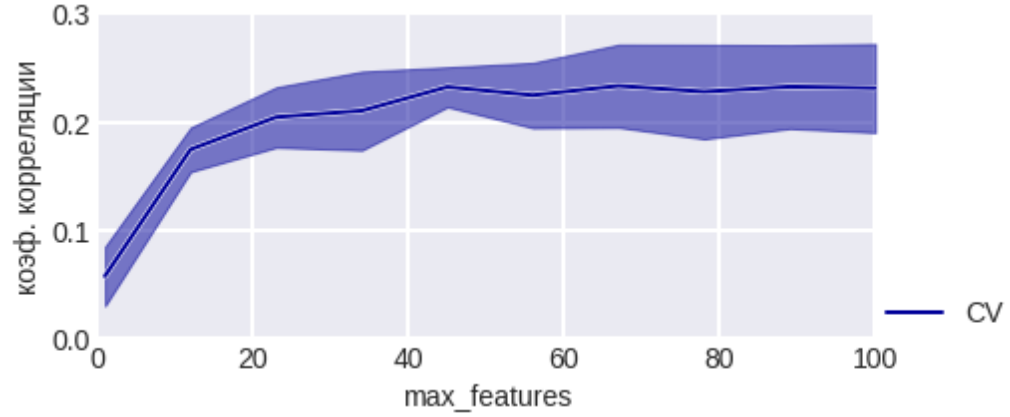

Рис. XX.05. Средний коэффициент корреляции между ответами деревьев леса.

На рис. XX.06 можно также видеть, что увеличение гиперпараметра max_features позволяет усложнить геометрию разделения: появляются области, окружающие объекты – «острова» и «заливы». Это связано с тем, что в данном случае при $\text{max\_features} = 1$ в каждом расщеплении выбирается случайный признак, а при $\text{max\_features} = 2$ просматриваются все признаки и находится наилучшее расщепление.

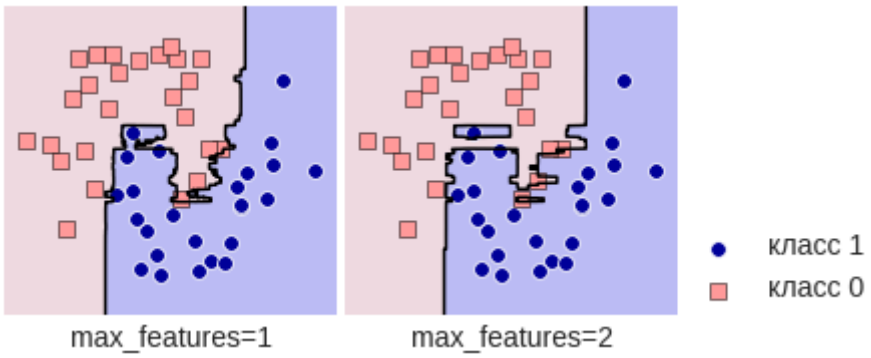

Рис. XX.06. Решение модельной задачи при разном числе признаков для выбора расщепления.

3. Максимальная глубина деревьев – max_depth

Чем меньше глубина, тем быстрее строится и работает случайный лес. При увеличении глубины возрастает качество на обучении (т.к. каждое дерево лучше настраивается на свою подвыборку), но и на контроле оно, как правило, увеличивается, см. рис. XX.07. Рекомендуется использовать максимальную глубину (кроме случаев, когда объектов слишком много и получаются очень глубокие деревья, построение которых занимает значительное время, или есть довольно много выбросов). Это кажется контринтуитивным, но такую рекомендацию давал и автор случайного леса Брейман[1], также она содержится в помощи к пакету sklearn[2]. На рис. XX.08 видно, что увеличение глубины приводит к более сложной геометрии разделения, на всех предыдущих рис. мы не ограничивали глубину дерева.

Используйте в RF максимально глубокие деревья.

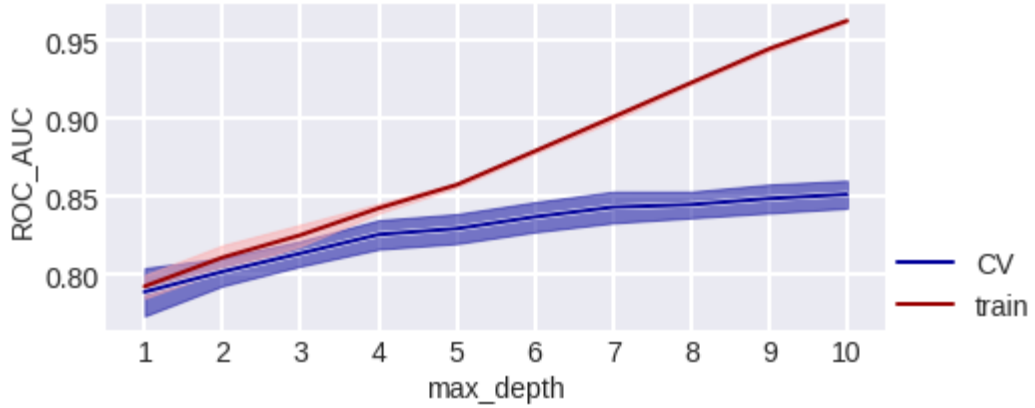

Рис. XX.07. Зависимость качества RF от максимальной глубины базовых деревьев[3].

4. Ограничение на число объектов в листьях – min_samples_leaf

Этот гиперпараметр регулирует, сколько объектов обучающей выборки каждого дерева может попадать в лист. Как правило, он не очень важный и можно оставить значение по умолчанию: авторы случайного леса предлагали использовать значение 5 в регрессии и 1 – в классификации (т.е. в классификации деревья «строятся до исчерпания выборки»)[4]. На рис. XX.09 видно, что на тесте качество не сильно меняется при изменении

[1] Breiman L. Random forests //Machine learning. – 2001. – Т. 45. – С. 5-32.

[2] Good results are often achieved when setting `max_depth=None` in combination with `min_samples_split=1` (i.e., when fully developing the trees). см. https://scikit-learn.sourceforge.net/stable/modules/ensemble.html

[3] На этом рисунке (и некоторых ниже) представлена зависимость качества решения реальной задачи определения пола пользователя по транзакциям от значений гиперпараметра.

[4] В библиотеке randomForest для R так и реализовано, в sklearn значение этого параметра по умолчанию – 1.

min_samples_leaf. Если этот гиперпараметр сделать очень большим, то будут получаться неглубокие деревья (он как и гиперпараметр max_depth влияет на сложность дерева). При использовании неглубоких деревьев (значение max_depth небольшое) изменение параметра min_samples_leaf не приводит к значимому эффекту, т.к. листья и так получаются «большими». На рис. XX.10 видна типичная картина зависимости качества от значения min_samples_leaf: она унимодальная.

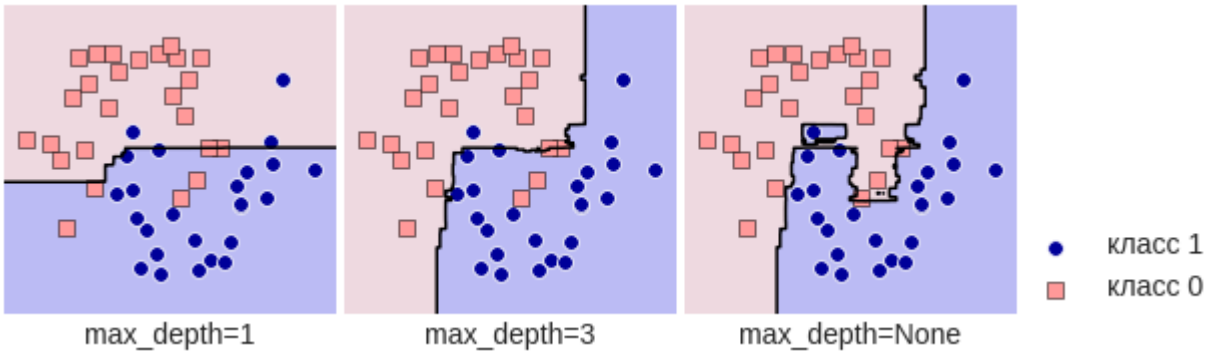

Рис. XX.08. Разделяющая поверхность RF при разной максимальной глубине.

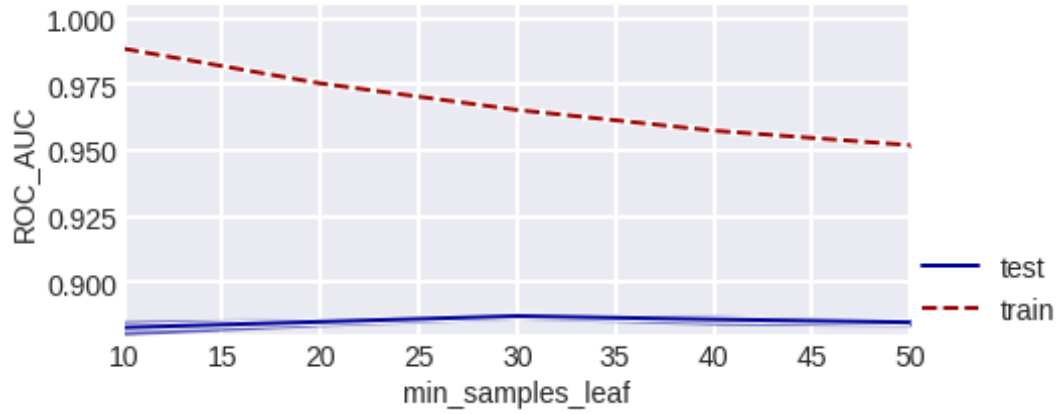

Рис. XX.09. Зависимость качества RF от ограничения на число объектов в листьях.

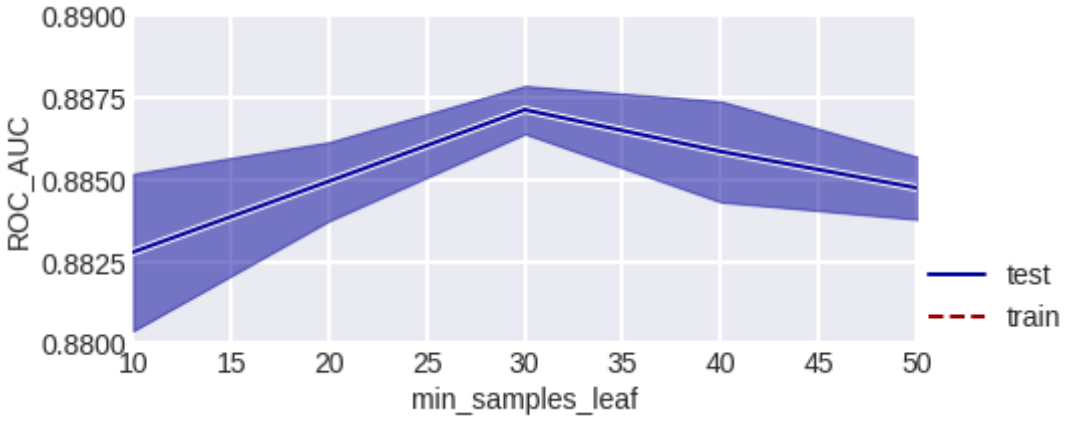

Рис. XX.10. Зависимость качества RF от ограничения на число объектов в листьях (увеличенный график).

5. Минимальное число объектов, при котором выполняется расщепление – min_samples_split

Все замечания, написанные про гиперпараметр min_samples_leaf, годятся и для этого гиперпараметра, который регулирует при каком числе объектов в области продолжается поиск расщепления в дереве, чтобы разбить её на подобласти. Графики качества также выглядят аналогично.

6. Критерий расщепления – criterion

Естественно ожидать, что это очень важный параметр, но в современных реализациях случайного леса не так уж и много критериев расщепления[1] и их выбор либо очевиден, либо не слишком заметно влияет на качество, см. рис. XX.11. Простой перебор поможет выбрать, что использовать в конкретной задаче.

7. Размер подвыборки для каждого дерева – max_samples

В классической реализации случайного леса использовалась процедура бутстрепа, т.е. каждое дерево обучалось на выборке, мощность которой совпадала с обучающей, но в ней были дубликаты. Чаще всего такая стратегия оптимальна. В современных реализациях есть возможность выбирать подвыборку без возвращения, а также выбирать подвыборку заданной мощности[2].

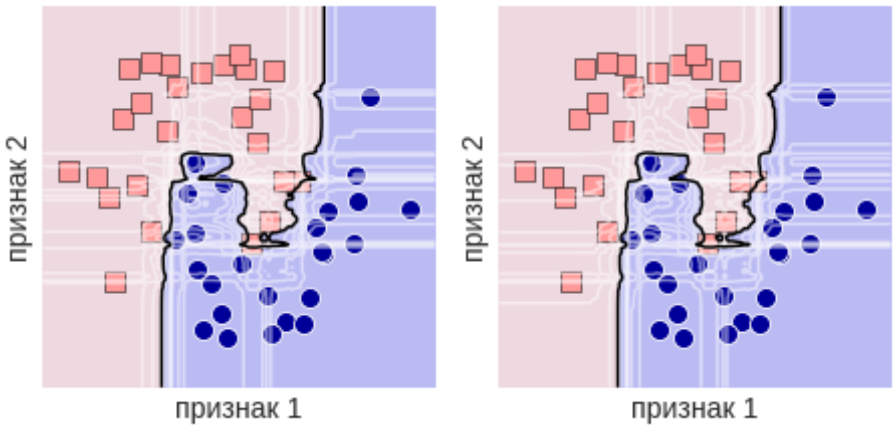

Рис. XX.11. Разделяющая поверхность RF при разных критериях расщепления: gini (слева), entropy (справа).

[1] В библиотеке sklearn для регрессии реализованы два критерия: «mse» и «mae», для классификации – «gini» и «entropy».

[2] Интересно, что в текущей версии sklearn (1.4.1) выбор *max_samples* влияет на обучение только в режиме бутстрепа.

Вариативность базисных алгоритмов в RF 1) за счёт бэгинга и 2) за счёт случайных подмножеств признаков для каждого расщепления.

Как и в любом бэгинге, в случайных лесах есть возможность получения OOB-ответов и OOB-ошибки. Также случайный лес позволяет **не просто получать ответ, но и оценивать уверенность в нём**. Например, в задаче регрессии среднее базовых алгоритмов – ответ, их дисперсия (или стандартное отклонение) – уверенность. Также есть возможность оценки важности признаков (feature_importances), для каждого признака это просто среднее важностей этого признака на отдельных деревьях. Заметим, что интуитивно у леса эта важность более адекватная (чем у отдельного дерева), т.к. он может зависеть от большего числа признаков.

Некоторые недостатки случайного леса наследуются из недостатков его базовых алгоритмов: деревьев. Например, с помощью этого ансамбля плохо решаются задачи экстраполяции, см. рис. XX.12 – за «пределами выборки» мы получаем константные решения.

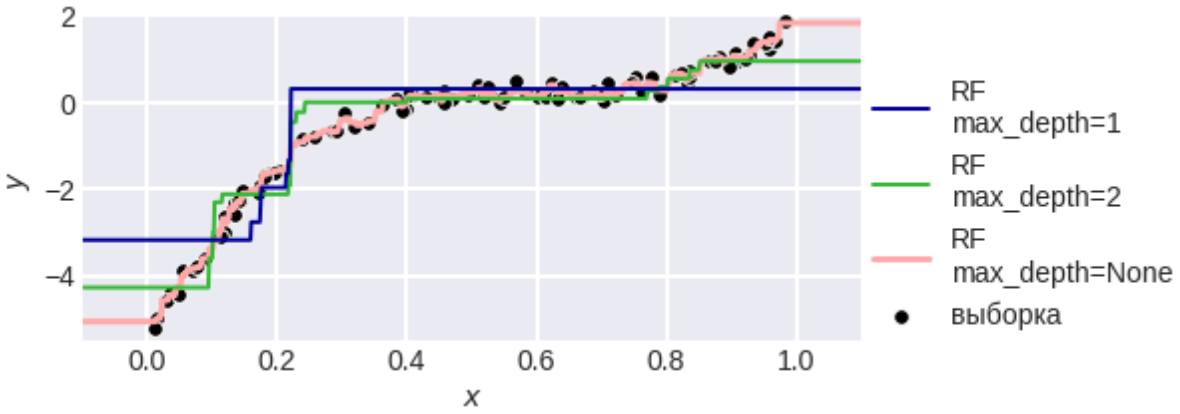

Рис. XX.12. Случайные леса в регрессии.

Другие варианты построения случайных лесов

Есть также модель родственная случайным лесам – **Extremely Randomized Trees (ExtraTrees[1])**. В классической реализации ExtraTrees нет бутстрепа: при построении каждого дерева используется вся выборка. Кроме того, если в случайных лесах для каждого рассматриваемого признака мы перебираем все возможные расщепления и выбираем наилучшее, то здесь мы выбираем лучшее расщепление по фиксированному числу случайных пар (признак, порог). Считается, что качество ExtraTrees сравнимо с качеством RF, кроме случаев

[1] Geurts P., Ernst D., Wehenkel L. Extremely randomized trees //Machine learning. – 2006. – Т. 63. – С. 3-42.

большого числа шумных признаков (здесь RF предпочтительнее). Зато ExtraTrees существенно быстрее обучается (строится ансамбль деревьев).

В методе **синтетический случайный лес (Synthetic RF[1])** строится стекинг лесов с разными значениями min_samples_leaf (в качестве мета-алгоритма также выбирается случайный лес).

Есть т.н. VR-деревья[2] (variable-random), в которых при формировании каждого узла с небольшой вероятностью выбирают случайный признак и случайное расщепление по нему.

Случайный лес: приложения

1. Опишем применение случайного леса в задаче «Search Results Relevance[3]», в которой по запросу и описанию выдачи (названию продукта и его подробному описанию) необходимо было определять релевантность продукта запросу. В обучающей выборке целевые значения были получены с помощью опроса асессоров, которые проставляли целые баллы от 1 (не релевантно) до 4 (релевантно). Признаки описывали сходства текстовых фрагментов, например, запроса и названия товара или описания товара и описания релевантного товара с таким же запросом; сходства измерялись разными способами (косинусное сходство, коэффициент Жаккара и т.п.), в том числе, с помощью различных нормировок (т.к. понятие «похоже» контекстное, надо учитывать похожесть на все описания и на основе распределения этих похожестей принимать решение). После формирования признакового пространства оказалось, что наиболее приемлемая для решения задачи модель – случайный лес. Но на практике, **случайный лес может быть плохо откалиброванным**. Здесь же при решении была следующая проблема: решалась задача регрессии с метками 1, 2, 3, 4. Если округлять полученный ответ (из-за усреднения деревьев он мог быть нецелым), то получается, что распределение ответов не похоже на распределение меток тестовой выборки. Для выравнивания распределений можно, например, подбирать пороги округления $-\infty = \theta_0 \le \theta_1 \le \theta_2 \le \theta_3 \le \theta_4 = +\infty$ в решающем правиле dr (decision rule), которое корректирует ответы алгоритма:

[1] Ishwaran H, Malley JD. «Synthetic learning machines». BioData Min. 2014;7(1):28 // https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4279689/pdf/13040_2014_Article_28.pdf

[2] Liu F. T. et al. Spectrum of variable-random trees //Journal of Artificial Intelligence Research. – 2008. – Т. 32. – С. 355-384.

[3] https://www.kaggle.com/c/crowdflower-search-relevance Описываем решение, которое вошло в топ-10 соревнования.

$$\mathrm{dr}(a(x)) = i \Leftrightarrow a(x) \in [\theta_{i-1}, \theta_i).$$

2. С помощью случайных лесов **можно строить специальную метрику** на объектах выборки. Каждое дерево индуцирует свою метрику на объектах выборки: расстояние по дереву между листьями, в которые попали объекты. Если дерево неглубокое и листья «достаточно большие», то можно даже использовать более простую метрику: она равна 0 для пары объектов из одного листа и 1 для пары объектов из разных листьев. Если теперь просуммировать такие метрики по всем деревьям, то получим метрику на множестве объектов, индуцированную лесом (расстояние между объектами равно числу деревьев леса, в которых эта пара объектов попадала в разные листья). Заметим, что она не зависит от масштаба признаков и может быть использована в метрических алгоритмах машинного обучения. К сожалению, мы в явном виде строим $m \times m$-матрицу попарных расстояний, что при большом числе объектов m затруднительно.

Задачи и вопросы

1. В данной главе в качестве модельной задачи для иллюстраций использовалась задача «два полумесяца». Может ли в ней быть следующая парадоксальная зависимость качества от параметра max_samples, см. рис. XX.13: чем меньше размер подвыборки для обучения, тем выше качество на скользящем контроле?

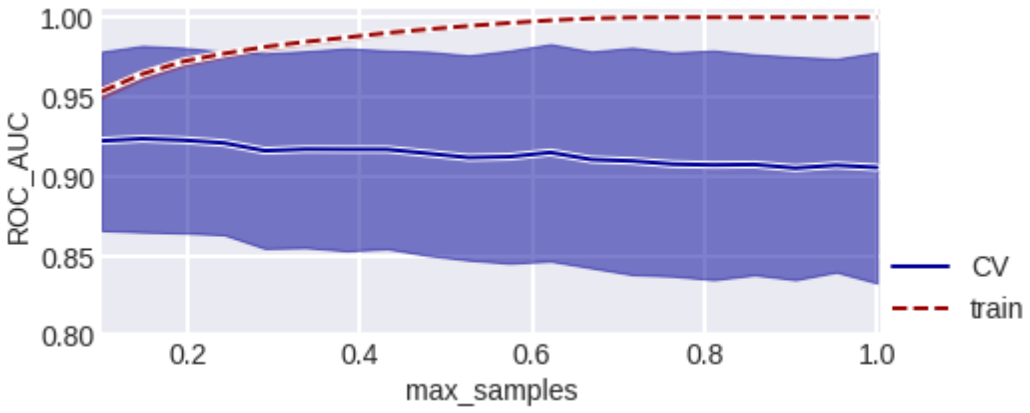

Рис. XX.13. Зависимость качества от размера подвыборки.

2. Было отмечено, что случайные леса позволяют получить не просто ответ, но и уверенность в нём. Как эту уверенность получить для задачи классификации?

Спасибо за внимание к книге!
Замечания по содержанию, замеченные ошибки
и неточности можно написать в телеграм-чате
https://t.me/Dyakonovsbook